ARM Processor Cortex™-A12 MPCore

Product Revision r0

Software Developers Errata Notice

Non-Confidential - Released

ARM

Software Developers Errata Notice

Non-Confidential Proprietary Notice

Web Address

http://www.arm.com

Feedback on content

If you have any comments on content, then send an e-mail to errata@arm.com . Give:

- the document title
- the document number, ARM-EPM-047351
- the page numbers to which your comments apply
- a concise explanation of your comments.

ARM also welcomes general suggestions for additions and improvements.

Release Information

Errata are listed in this section if they are new to the document, or marked as "updated" if there has been any change to the erratum text in Chapter 2. Fixed errata are not shown as updated unless the erratum text has changed. The summary table in section 2.2 identifies errata that have been fixed in each product revision.

18 Jul 2013: Changes in Document v1

No new or updated errata in this document version.

17 Oct 2013: Changes in Document v2

Page	Status	ID	Cat	Rare	Summary of Erratum
8	New	812078	CatA		A snoop request sent to a CPU just going out of reset might lead to a deadlock
10	New	812076	CatB		Potential deadlock due to CMO / MRC / conditional load / TLB maintenance combination
15	New	812077	CatB	Rare	Barriers might not behave properly with respect to a Non Cached store
18	New	812075	CatC		Processor IDs unavailable through the APB bus

13 Nov 2013: Changes in Document v3

Page	Status	ID	Cat	Rare	Summary of Erratum
9	New	814276	CatA		CMO in L2 Cache might not respect ordering with natural evictions
13	New	814277	CatB		Possible data corruption due to a CMO targeting a CPU in retention state.
22	New	814278	CatC		Execution of an UNPREDICTABLE POP instruction may deadlock
23	New	815722	CatC		A cacheable store might not be visible if a previous write on the same cache line took an external abort
24	New	815723	CatC		Potential data corruption out of reset

16 Dec 2013: Changes in Document v4

Page	Status	ID	Cat	Rare	Summary of Erratum
9	Updated	814276	CatA		A Data Cache Maintenance Operation by MVA might not behave with regards to a non coherent agent.
10	New	818322	CatA		A TLBIALLIS or TLBIALLNSNHIS operation might fail
12	New	814273	CatB		An access may get initiated on the L2 TAG RAMs while they are still in retention mode.
14	New	817124	CatB		Reset state of ATB Flush acknowledge is incorrect
15	New	818327	CatB		A virtual interrupt asserted on vIRQ does not wake up a CPU in retention state
15	New	819320	CatB		TLBIMVAA/TLBIMVAAIS may fail to invalidate all targeted entries
17	New	818324	CatB	Rare	TLB and Icache maintenance operations might not be properly broadcast when HCR.FB is set
18	New	818325	CatB	Rare	Execution of an UNPREDICTABLE STR or STM instruction might deadlock
20	New	814274	CatC		PTM: external timestamp request or OS Lock de-assertion event lost when a CPU is in WFI Status
21	New	814275	CatC		PTM: Possible stall of the debugger when trying to flush ATB bus of a processor in WFI
25	New	818319	CatC		HSR.ISS.SRT field not correctly updated on Stage2 Abort on T2EE-specific STR instruction
26	New	818320	CatC		SCUCTLR processor power status field is erroneous.
27	New	818321	CatC		LDM (Exception Return) with write-back instructions are counted twice in exception return event (0xa)
28	New	818323	CatC		Exception Taken event does not reflect the real number of exceptions taken when using event filtering
29	New	818326	CatC		Execution of an UNDEF CPS-like instruction in debug state might deadlock
30	New	818969	CatC		Read-after-read ordering might not be properly ensured for transfers longer than a cacheline
31	New	819319	CatC		Debugger might stall

Contents

Chapter 1.
Introduction

This chapter introduces the errata notice for the ARM Cortex-A12 MPCore processor.

1.1. Scope of this document

This document describes errata categorized by level of severity. Each description includes:

- the current status of the defect
- where the implementation deviates from the specification and the conditions under which erroneous behavior occurs
- the implications of the erratum with respect to typical applications
- the application and limitations of a 'work-around' where possible

This document describes errata that may impact anyone who is developing software that will run on implementations of this ARM product.

1.2. Categorization of errata

Errata recorded in this document are split into the following levels of severity:

Table 1 Categorization of errata

Errata Type	Definition
Category A	A critical error. No workaround is available or workarounds are impactful. The error is likely to be common for many systems and applications.
Category A(rare)	A critical error. No workaround is available or workarounds are impactful. The error is likely to be rare for most systems and applications. Rare is determined by analysis, verification and usage.
Category B	A significant error or a critical error with an acceptable workaround. The error is likely to be common for many systems and applications.
Category B(rare)	A significant error or a critical error with an acceptable workaround. The error is likely to be rare for most systems and applications. Rare is determined by analysis, verification and usage.
Category C	A minor error.

Chapter 2.
Errata Descriptions

2.1. Product Revision Status

The r*n*p*n* identifier indicates the revision status of the product described in this book, where:

r*n* Identifies the major revision of the product.

p*n* Identifies the minor revision or modification status of the product.

2.2. Revisions Affected

Table 2 below lists the product revisions affected by each erratum. A cell marked with **X** indicates that the erratum affects the revision shown at the top of that column.

This document includes errata that affect revision r0 only.

Refer to the reference material supplied with your product to identify the revision of the IP.

Table 2 Revisions Affected

ID	Cat	Rare	Summary of Erratum	r0p0	r0p1
818322	CatA		A TLBIALLIS or TLBIALLNSNHIS operation might fail	X	
814276	CatA		A Data Cache Maintenance Operation by MVA might not behave with regards to a non coherent agent.	X	
812078	CatA		A snoop request sent to a CPU just going out of reset might lead to a deadlock	X	
818327	CatB		A virtual interrupt asserted on vIRQ does not wake up a CPU in retention state	X	
817124	CatB		Reset state of ATB Flush acknowledge is incorrect	X	
814277	CatB		Possible data corruption due to a CMO targeting a CPU in retention state.	X	
814273	CatB		An access may get initiated on the L2 TAG RAMs while they are still in retention mode.	X	
812076	CatB		Potential deadlock due to CMO / MRC / conditional load / TLB maintenance combination	X	
818325	CatB	Rare	Execution of an UNPREDICTABLE STR or STM instruction might deadlock	X	
818324	CatB	Rare	TLB and Icache maintenance operations might not be properly broadcast when HCR.FB is set	X	
812077	CatB	Rare	Barriers might not behave properly with respect to a Non Cached store	X	
819319	CatC		Debugger might stall	X	
818969	CatC		Read-after-read ordering might not be properly ensured for transfers longer than a cacheline	X	X
818326	CatC		Execution of an UNDEF CPS-like instruction in debug state might deadlock	X	
818323	CatC		Exception Taken event does not reflect the real number of exceptions taken when using event filtering	X	

ID	Cat	Rare	Summary of Erratum	r0p0	r0p1
818321	CatC		LDM (Exception Return) with write-back instructions are counted twice in exception return event (0xa)	X	
818320	CatC		SCUCTLR processor power status field is erroneous.	X	
818319	CatC		HSR.ISS.SRT field not correctly updated on Stage2 Abort on T2EE-specific STR instruction	X	
815723	CatC		Potential data corruption out of reset	X	
815722	CatC		A cacheable store might not be visible if a previous write on the same cache line took an external abort	X	
814278	CatC		Execution of an UNPREDICTABLE POP instruction may deadlock	X	
814275	CatC		PTM: Possible stall of the debugger when trying to flush ATB bus of a processor in WFI	X	
814274	CatC		PTM: external timestamp request or OS Lock de-assertion event lost when a CPU is in WFI Status	X	
812075	CatC		Processor IDs unavailable through the APB bus	X	

2.3. Category A

812078: A snoop request sent to a CPU just going out of reset might lead to a deadlock

Category A

Products Affected: Cortex-A12 MPCore.

Present in: r0p0

Description

A CPUn that receives a snoop request while leaving reset, might lose the snoop request and deadlock.

Configurations affected

This erratum affects all configurations of the Cortex-A12 with at least two coherent agents(MP1+ACP, or two or more CPUs).

Conditions

This erratum occurs when:

- A CPUn has data within its data cache and, at the time of entering reset, has not replaced the cache line as the result of natural replacement.

- A CPUm or the ACP is performing an access to the same cache line, which triggers a snoop request to CPUn in the same cycle when CPUn is leaving reset.

- While CPUn was in reset state, neither a CPUm nor the ACP has performed an access to the same cache line, which would have triggered a snoop request to CPUn.

Note:

- There is a single clock cycle window during which the conditions are met that will trigger this erratum.

- Cleaning and invalidating the L1 cache before going into reset on CPUn does not prevent the erratum.

Implications

This erratum might cause a system deadlock.

Workaround

There is no workaround for this erratum.

814276: A Data Cache Maintenance Operation by MVA might not behave with regards to a non coherent agent.

Category A

Products Affected: Cortex-A12 MPCore.

Present in: r0p0

Description

A CPU executing a Data Cache Maintenance Operation by MVA might fail to ensure that the effect of this operation is visible to a non coherent agent.

Configurations Affected

All configurations of Cortex-A12 are affected.

Conditions

- A CPU doing a Data Cache Maintenance Operation by MVA at address A.

- A non coherent agent, relying on the effect of the Data Cache Maintenance Operation, and performing action on the external memory at address A.

Implications

The Data Cache Maintenance Operation by MVA might not evict the line from the cluster, or might fail to invalidate the cache line within the cluster. As such:

- A non coherent agent might read a stale value instead of the evicted data still present in the cluster

- A CPU within the cluster might read stale value if the memory has been written by a non coherent agent.

Note that only non coherent agents are affected by this errata. No access to the same address A performed by either of a CPU in the cluster, or through the ACP port, is affected by this errata.

Workaround

There is no workaround.

818322: A TLBIALLIS or TLBIALLNSNHIS operation might fail

Category A
Products Affected: Cortex-A12 MPCore.
Present in: r0p0

Description

CPUs executing in Non-secure PL0/PL1 and receiving a broadcast TLBIALLIS or TLBIALLNSNHIS operation targeting both Stage1 and Stage2 translations might fail to invalidate all targeted Stage2 translations. As a result, affected CPUs might continue to use stale Stage2 translations after the invalidation operation is complete.

Configurations affected

This erratum affects Cortex-A12 configurations with more than one CPU.

Conditions

CPUs receiving the broadcast operation must be executing in Non-secure PL0/1 mode with the HCR.VM bit set. In addition, this erratum only occurs if the invalidated Stage2 translations pointed to physical memory holding Stage1 translation tables currently in use by the CPU receiving the broadcast operation.

Implications

The affected CPUs might continue to use the invalidated Stage2 translations after the invalidation was complete. This might cause incorrect address translation by the MMU, leading to data corruption.

Workaround

The use of an Inter-Processor Interrupt to perform the required invalidation operation prevents this erratum from occurring.

2.4. Category A (Rare)

There are no errata in this category

2.5. Category B

812076: Potential deadlock due to CMO / MRC / conditional load / TLB maintenance combination

Category B
Products Affected: Cortex-A12 MPCore.
Present in: r0p0

Description

An inter-dependency between all of the following might cause a deadlock to occur under rare circumstances:

- Either of an Icache maintenance, a TLB maintenance, or a DSB broadcast operation sent by CPUn.
- A CP15 read on CPUn.
- A conditional load, or a store, with a dependency on the CP15 read on CPUn.
- A broadcast TLB maintenance operation received from another CPU to CPUn.
- A broadcast DSB received from another CPU to CPUn.

Configurations affected

This erratum affects MP configurations of Cortex A12 containing two or more processors.

Conditions

Under rare circumstances, the following code sequence might cause a deadlock:

CPUn :

- An Icache maintenance, or a TLB maintenance, or a DSB, being broadcast (A)

- MRC Rx, X (B)

- A conditional load, or a store, having a dependency on the MRC instruction (C)

Other CPUs:

- Broadcast TLB maintenance operation (D)

- Broadcast DSB (E)

Notes:

- (D) and (E) can be sent by two different CPUs.

- For the store (C), the dependency on the MRC instruction (C) is any of a data dependency, a condition code depending of the result of the data read by the MRC, or being in program order after a conditional branch which condition code depends on the result of the data read by the MRC.

This results in the following situation in CPUn:

1. The conditional load or the store (C) is blocked in CPUn waiting on the condition code resolution or on the data that depends on 2.
2. The CP15 read (B) that cannot progress due to 3.
3. The pending broadcast operation (A) from CPUn to CPUm not getting its response due to 4.
4. The broadcast DSB (E) received from CPUm waiting for 5.
5. The conditional load or the store (C) in 1 being stalled in the Load Store Unit.

The deadlock is seen under specific timing conditions: the broadcast DSB must be received when the conditional load or the store is already present in Load Store Unit and the CP15 read operation has not yet been accepted by the CP15 unit.

Implications

The erratum might lead to system deadlock.

Workaround

Inserting a DSB or an ISB after the CP15 read (B) avoids the deadlock.

814273: An access may get initiated on the L2 TAG RAMs while they are still in retention mode.

Category B

Products Affected: Cortex-A12 MPCore.

Present in: r0p0

Description

A CPU leaving low power state (either of power down or WFI), or an access on the ACP, can initiate an access to the L2 TAG RAMs while these are still in retention mode, leading to possible corruption.

When one of the CPU leaves a low power state, or when the ACP is enabled, L2QACTIVE signals the power controller that the L2 RAMs' power domain must leave the retention state. The power controller should power up the RAMs, and then complete the L2Qchannel handshake. Until the Qchannel handshake is completed, no access should be made to the L2 RAMs. Unfortunately, nothing prevents an access from happening.

Configuration affected

Requires the implementation of L2 retention.

Conditions

The L2 TAG RAMs are in retention state. The CPU leaving the low power mode, or an access made on the ACP, will trigger an L2QACTIVE request to the power controller to leave the retention mode. The request on the ACP might trigger an L2 TAG lookup before the L2 QChannel handshake is complete.

Implications

A data corruption might occur.

Workaround

The L2QChannel must be disabled : bit[0] of the L2ECTLR must be set to 1.

Assembly language sequence :

MRC p15, 1, Rx, c9, c0, 3

ORR Rx,Rx,0x00000001

MCR p15, 1, Rx, c9, c0, 3

814277: Possible data corruption due to a CMO targeting a CPU in retention state.

Category B

Products Affected: Cortex-A12 MPCore.

Present in: r0p0

Description

A CPU put into retention state does not receive Cache Maintenance Operations from other processors, leading to potential data corruption.

This happens even if the ACTLR.SMP bit is high.

Configuration affected

- CPU retention implemented.
- MP configuration.

Conditions

#1 CPUn is in retention state.

#2 CPUm broadcast an Instruction Cache or TLB operation that should affect the state of CPUn.

Implications

Instruction cache and TLB maintenance is not performed, which might lead to execution of obsolete instructions or data corruption.

Workaround

Disable CPU qchannels: Set bit 18, 22, 26, 30 of the SCUCTLR register to disable CPU 0,1,2,3 qchannels respectively.

Assembly language sequence:

MRC p15, 1, Rx, c9, c0, 4

ORR Rx,Rx,0x44440000

MCR p15, 1, Rx, c9, c0, 4

Or flush the Icache and/or TLB before entering retention state if needed.

817124: Reset state of ATB Flush acknowledge is incorrect

Category B

Products Affected: Cortex-A12 MPCore.

Present in: r0p0

Description

The ATB protocol includes a flush request that can be generated by a trace sink, and is then distributed to all of the connected trace sources. The flush channel consists of a valid/ready handshake. As a result of this erratum, when the Cortex-A12 debug domain is held in reset, the processor will not respond to a flush request.

Conditions

- A system where one of the CPU can be held in reset, or power gated with the AFREADYM output clamped low.
- The CPU ATB master interface is connected to a component in a different reset domain.
- A component connected to the ATB master interface requests a flush of a CPU that is held in reset or low power state.

Implications

The flush request remains pending until the affected CPU is released from reset and receives a clock. Any other trace sources in the system remain able to generate trace, but the component that requested a flush will behave as if the flush is still in progress. This might prevent any components that are downstream of the affected CPU from entering a low power state.

Workaround

Clamp AFREADYM HIGH when it is driven by a CPU that is power-gated.

AFREADYM should be driven HIGH when it is driven by a CPU that is held in reset.

818327: A virtual interrupt asserted on vIRQ does not wake up a CPU in retention state

Category B
Products Affected: Cortex-A12 MPCore.
Present in: r0p0

Description

A virtual IRQ/FIQ might not wake up a CPU from retention state, regardless of HCR.IMO and HCR.FMO state.

Configuration affected

Implementations that make use of CPU retention are affected.

Conditions

CPU is in retention state. A virtual IRQ/FIQ targeting the CPU in retention state is asserted.

Implications

The CPU remains in retention.

Workaround

Ensure the CPU is taken out of retention by a method other than virtual interrupts.

or

Disable CPU qchannels:

MRC p15, 1, Rx, c9, c0, 4

ORR Rx,Rx,0x44440000

MCR p15, 1, Rx, c9, c0, 4

2.6. Category B (Rare)

812077: Barriers might not behave properly with respect to a Non Cached store

Category B Rare
Products Affected: Cortex-A12 MPCore.
Present in: r0p0

Description

Under very rare circumstances, on a CPU executing a non-cached (non-cacheable normal memory or

strongly ordered or device) store followed by a barrier (DMB or DSB), the DMB might not ensure

proper ordering between the non-cached store and subsequent cached accesses, or the DSB might not ensure proper completion of the non-cached store access.

Configurations affected

This erratum affects all configurations of the Cortex-A12

Conditions

- CPUn executes a non-cached store followed by a DMB or DSB.
- CPUm or any external observer is reading the same location.

Implications

Barriers might not behave properly after a store to a non-cached location, which might cause a potential data corruption in cases where another agent is reading the same location. Under very rare circumstances CPUm or the external observer might read a stale value instead of the data written by the non-cached store.

Workaround

CPUn doing a non cached read after the barrier removes the erratum. Note that the added read might be to any non cached memory location, and does not need to be to the same memory location as the non cached store.

818324: TLB and Icache maintenance operations might not be properly broadcast when HCR.FB is set

Category B Rare

Products Affected: Cortex-A12 MPCore.

Present in: r0p0

Description

When a CPU executing in Non-Secure PL0/PL1 with the HCR.FB bit set tries to execute a TLB or Icache maintenance operation, this operation might not be properly broadcast to all CPUs, resulting in stale TLB and Icache entries. This happens regardless of whether HCR.FB would have had an effect on the operation being executed: all TLBI* and ICI* operations are affected by this erratum, including their Inner-Shared variants.

Configurations affected

This erratum affects Cortex-A12 configurations with more than one CPU.

Conditions

- Hypervisor sets HCR.FB.

- Non-Secure PL0/PL1 executes a TLB or Icache maintenance operation. This operation is correctly broadcast.

- Non-Secure PL0/PL1 executes another TLB or Icache maintenance operations. This operation and subsequent TLB/Icache operations will not be broadcast.

The erratum does not occur if between two affected operations, the CPU either:

- takes an exception to Secure PL0/1 or PL2

- performs a CP15 write operation that is not an affected operation.

Implications

The consequences of stale TLB and Icache entries are unpredictable and might cause spurious TLB faults, data corruption or incorrect opcodes being fetched and executed.

Workaround

There are three possible workarounds:

- force the guest OS to run on only a defined physical CPU.

or

- trap TLBI and ICI operations and perform them from PL2.

or

- Force a clean and Invalidate of both the Icache and the TLB before migrating the OS from one physical CPU to another physical CPU.

818325: Execution of an UNPREDICTABLE STR or STM instruction might deadlock

Category B Rare

Products Affected: Cortex-A12 MPCore.

Present in: r0p0

Description

When a CPU executes a sequence of two conditional store instructions with opposite condition code and updating the same register, the system might enter a deadlock if the second conditional instruction is an UNPREDICTABLE STR or STM instruction.

Configurations affected

This erratum affects all Cortex-A12 configurations.

Conditions

- A sequence of two2 STR or STM (maximum two2 registers in the list) with write-back instructions is executed.

- Both instructions are conditional and have opposite condition codes (for example EQ and NE).

- Both instructions update the same base register.

- The second instruction is an UNPREDICTABLE STR or STM (maximum two2 registers in the list) with write-back and the write-back register is in the list of stored registers.

Implications

The system might deadlock.

Workaround

Setting bit[12] of the Feature Register prevents the erratum.

```
MRC p15, 0, r0, c15, c0, 1
ORR r0, r0, #1<<12
MCR p15, 0, r0, c15, c0, 1
```

2.7. Category C

812075: Processor IDs unavailable through the APB bus

Category C

Products Affected: Cortex-A12 MPCore.

Present in: r0p0

Description

The Debug Management Processor IDs cannot be accessed as they should be through the debug APB bus under the following conditions.

- They cannot be accessed through the Memory Mapped debug interface when the CORE is powered down or when the debug registers have been locked by the DBGOSDLR.DLK

- They are not available through the External Debug Interface when the debug registers have been locked by the DBGOSLSR.OSLK.

Configurations affected

This erratum affects all configurations of the processor.

Conditions

This erratum is visible under any of the following conditions:

- The DBGOSDLR.DLK is set, OR
- The DBGOSLSR.OSLK is set, OR
- The core is in power down.

Implications

Under the conditions described above, an access through the External Debug Interface on the APB bus will return a PSLVERR error. This causes the topology detection to fail.

Workaround

The external debugger must use an external configuration.

814274: PTM: external timestamp request or OS Lock de-assertion event lost when a CPU is in WFI Status

Category C

Products Affected: Cortex-A12 MPCore.

Present in: r0p0

Description

The PTM might lose synchronization events when the core is in WFI / WFE state and the PTM has drained all data and entered the IDLE state.

In such a situation the PTM clock is stopped; an external timestamp request or setting the OS Lock to zero might be ignored by PTM.

Configurations Affected

All implementations are affected.

Conditions

1.PTM is tracing.

2.The core is in WFI or WFE state and the PTM has drained all data and entered the IDLE state.

3.The OS Lock is set to zero or Timestamp request is set.

Implications

The PTM might not generate:

1) the expected A-SYNC packet when the OS Lock is set to zero,

- although the expected I-Synch and T-Sync packets will be generated

2) a Timestamp packet when a timestamp request occurs.

Note: When the OS Lock is cleared during WFI/WFE state, the I-sync and T-sync are generated properly.

Workaround

1) Unset the OS Lock or request timestamp when the core is not in WFI.

2) Synchronize the trace on an ISYNC packet.

or

3) Prevent the PTM to be clock gated while the processor is in WFI

- Disable CPU qchannels :set bit 18, 22, 26, 30 of the SCUCTLR register to disable CPU 0,1,2,3 qchannels respectively.

- Disable dynamic PTM clock gating : set bit 8 of the feature register.

Assembly language sequence :

```
; Disable PTM clock gating
MRC p15, 0, Rx, c15, c0, 2
ORR.W R1,R1,#1<<8
MCR p15, 0, Rx, c15, c0, 2

; Disable CPU qchannels
MRC p15, 1, Rx, c9, c0, 4
ORR Rx,Rx,0x44440000
MCR p15, 1, Rx, c9, c0, 4
```

814275: PTM: Possible stall of the debugger when trying to flush ATB bus of a processor in WFI

Category C

Products Affected: Cortex-A12 MPCore.

Present in: r0p0

Description

The AFVALID signal to the PTM that all buffers must be flushed because trace capture is about to stop. The PTM asserts AFREADY to indicate that buffers have been flushed.

While in WFI state, the flush request AFVALID might not get acknowledged until the CPU is woken up.

Configurations Affected

All configurations are affected.

Conditions

1. QChannels enabled.

2. The core is in WFI state and the PTM has drained all data and entered in IDLE state.

3. AFVALID Flush request is sent.

Implications

If the above conditions occur, the PTM might not respond to the ATB flush requests, leading to a stall of the debugger.

Workaround

1) Disable CPU qchannels :set bit 18, 22, 26, 30 of the SCUCTLR register to disable CPU 0,1,2,3 qchannels respectively.

Assembly language sequence :

MRC p15, 1, Rx, c9, c0, 4

ORR Rx,Rx,0x44440000

MCR p15, 1, Rx, c9, c0, 4

or

2) Do not request an ATB flush while the processor is in WFI.

814278: Execution of an UNPREDICTABLE POP instruction may deadlock

Category C

Products Affected: Cortex-A12 MPCore.

Present in: r0p0

Description

When a CPU executes a POP instruction which is UNPREDICTABLE because PC and R13 are included in its register list, the system might enter a deadlock state which it will only leave on reception of an external interrupt or an external abort.

Configurations affected

This erratum affects all Cortex-A12 configurations.

Conditions

- The POP instruction must have PC and R13 in its register list.
- The POP instruction must be received by the decoder with a valid branch prediction from fetch unit.

Implications

The system might enter a deadlock state which it will only leave on reception of an external interrupt or an external abort.

Workaround

An external interrupt, or an external abort, will exit the deadlock and the exception will be serviced.

815722: A cacheable store might not be visible if a previous write on the same cache line took an external abort

Category C

Products Affected: Cortex-A12 MPCore.

Present in: r0p0

Description

When a cacheable store is taking an external abort on its linefill, the line is allocated as invalid (allocation is done with valid bit set to 0 in L1 tagRAM). Later, if another store is done on the same cache line, it might reuse the cache way information of the previous store, ignoring the abort. In such a case the store might not do a lookup/linefill and might behave as if the line was really present in Data cache: the write is done in cache on a line that is still invalid meaning that the write is lost.

The same scenario might occur several times until:

- The corresponding index/way is naturally replaced in Data Cache.

- Several stores to different addresses are done, filling the store buffer that will not be able to reuse the aborting store's way information anymore.

Note that relying on external aborts to protect the main memory subsystem has severe limitations.

As a consequence, this defect is not expected to cause any issue in practice because this protection scheme is not expected to be implemented in real systems.

Configurations affected

All Cortex-A12 configurations

Conditions

To be visible the erratum requires:

1) Cache line A being read protected by an external abort.

2) A cacheable store in cache line A generating a linefill.

3) Cache line A not being protected anymore by an external abort.

4) A cacheable store in cache line A.

Implications

Data of Stores might be lost until the index/way is naturally replaced in cache or until enough stores to other cache lines have been executed to allow way information to be discarded in the processor.

As mentioned in the description section, the erratum is not expected to cause any significant harm in practice, because this external protection scheme is severely limited and unlikely to be implemented in real systems.

Workaround

There is no workaround for this erratum.

815723: Potential data corruption out of reset

Category C
Products Affected: Cortex-A12 MPCore.
Present in: r0p0

Description

Under rare circumstances, a cacheable Load operation performed shortly after CPU power-up, might receive erroneous cache hit information from the Store Buffer causing potential data corruption.

Configurations affected

This erratum affects all configurations of Cortex A12.

Conditions

The problem is only visible shortly after power up before Stores to at least 8 different 64-bit ranges have been performed and after the data cache has been turned on.

For each of the 8 Store Buffer slots, a one-cycle window exists after CPU power-up, during which a cacheable Load operation, targeting the same 64-bit address range as the store operation contained in this Store Buffer slot, might receive erroneous Store Buffer cache hit information leading to data corruption.

Implications

The erratum might cause data corruption.

Workaround

There is a possible workaround:

- After CPU power-up, the CPU could perform at least 8 Uncacheable stores to 8 distinct 64-bit regions before performing a DSB and turning on its data cache

818319: HSR.ISS.SRT field not correctly updated on Stage2 Abort on T2EE-specific STR instruction

Category C

Products Affected: Cortex-A12 MPCore.

Present in: r0p0

Description

For T2EE-specific STR immediate instruction (STR<c> <Rt>, R9, #<imm>), in cases where a Stage 2 data abort that reports a valid instruction syndrome in the HSR, the value of the HST.ISS.SRT field is wrong

Configurations affected

This erratum affects all configurations of Cortex A12.

Conditions

Performing a T2EE-specific STR immediate instruction (STR<c> <Rt>, R9, #<imm>) that will trigger a Stage 2 data abort on the the final stage of translation, the value of the HST.ISS.SRT field will always update with 0 instead of the register address

Implications

The erratum might cause reporting issues.

Workaround

There is a possible workaround:

- The hypervisor SW can try and get the exact opcode of the instruction from the ELR_hyp register when the instruction syndrome fields of the SPSR indicate that we have taken the abort from a Thumb2EE state, the Instruction syndrome in the HSR is valid, HSR.ISS.SRT is zero and the HSR.IL indicates a 16-bit instruction.

818320: SCUCTLR processor power status field is erroneous.

Category C

Products Affected: Cortex-A12 MPCore.

Present in: r0p0

Description

The processor's power status fields in the SCUCTLR register indicate that a CPU is in retention state when it is actually in power down.

Configuration affected

All configurations except single CPU builds are affected.

Conditions

The SCUCTLR is read when a CPU is in power down.

Implications

The processor power status field is incorrect.

Workaround

There is no workaround

818321: LDM (Exception Return) with write-back instructions are counted twice in exception return event (0xa)

Category C

Products Affected: Cortex-A12 MPCore.

Present in: r0p0

Description

Instructions LDM (Exception Return) with write-back are always counted twice in architectural exception return event 0xa

Configurations affected

This erratum affects all configurations of Cortex-A12.

Conditions

The erratum only occurs on instructions with a write-back of the base register.

Implications

Due to this erratum, Exception Return event 0xa may not reflect the real number of exception return instructions architecturally executed.

Workaround

A possible workaround can be for the software to count this event directly by instrumenting the code on Exception return instruction.

818323: Exception Taken event does not reflect the real number of exceptions taken when using event filtering

Category C

Products Affected: Cortex-A12 MPCore.

Present in: r0p0

Description

Exception Taken event (0x9) does not count exceptions on privileged level or security level changes when event filtering is used.

Configurations affected

This erratum affects all configurations of Cortex-A12.

Conditions

The erratum occurs when event filtering is used.

Implications

Due to this erratum, Exception Taken event 0x9 might not reflect the real number of exceptions taken when event filtering is used.

Workaround

A workaround for this erratum is to increment an software event counter when entering after receiving an Exception by using the SPSR to know the Mode from which we are coming.

818326: Execution of an UNDEF CPS-like instruction in debug state might deadlock

Category C

Products Affected: Cortex-A12 MPCore.

Present in: r0p0

Description

When the CPU is halted in debug state and executes an UNDEF CPS-like opcode through ITR, the system might deadlock.

Configurations affected

This erratum affects all Cortex-A12 configurations.

Conditions:

The processor is halted in debug state.

The opcode executed through ITR must be in ARM uncond encoding space with following restrictions:

- opcode[27:26]=00 and opcode[16]=0 (CPS-like instruction)
- opcode[17]=1 (instruction modifying cpu mode)
- opcode[25:20] different from 010000, and generating an UNDEF

Implications

The system might deadlock.

Workaround

Debugger can execute an ISB through ITR after deadlocking CPS-like opcode to release the CPU.

818969: Read-after-read ordering might not be properly ensured for transfers longer than a cacheline

Category C

Products Affected: Cortex-A12 MPCore.

Present in: r0p0, r0p1

Description

The load-store unit might fail to properly enforce read-after-read ordering between younger loads and older loads that read more than 64 bytes of data from memory.

Configurations affected

This erratum affects Cortex-A12 configurations with more than one CPU.

Conditions

The erratum only occurs for loads that read more than 64 bytes, which is only possible using NEON instructions. For the erratum to occur, the data must be modified by a CPU B between the time it observes the younger load from CPU A and the time it observes the older load from CPU A.

Implications

A younger Load might observe older data than an older load in program order. This erratum causes a violation of the memory ordering rules, resulting in possible data corruption.

Workaround

The programmer should not use instructions loading more than 64 bytes of data from memory when read-after-read ordering must be ensured.

819319: Debugger might stall

Category C

Products Affected: Cortex-A12 MPCore.

Present in: r0p0

Description

A debug restart request coming from the cross trigger interface, or from an apb access to DBGDRCR.RRQ might not make the DBGDSCR.Halted update visible to the external debugger.

The processor resumes normal execution, but continues to indicate through DBGDSCR.Halted that the debug state has not been left.

DBGACK and DBGRESTARTED will stay high and some of the DBGDSCR flags can be updated.

Configurations affected

All configurations of Cortex-A12 are affected.

Conditions

The errata occurs when the CTI RESTART Request is asserted before the Cortex-A12 is halted (DBGDSCR.Halted).

Implications

1.The processor will indicate that it is halted through DBGDSCR.Halted, even if the normal execution resumed.

2.DBGDSCR ADAbort, SDAbort and Und flags will be updated as if in debug state.

3.DBGACK and CTI DBG Restarted will remain high, possibly causing the external debugger to deadlock.

Workaround

If the Cortex-A12 is restarted using CTI RESTART Req the debugger must poll the DBDGSCR.Halted to be sure it is halted before restarting it and must not maintain this pin high.